Econ 172A - Slides from Lecture 15

Joel Sobel

November 27, 2012

Announcements

- Joel prepared this slide. Herb is presenting it.
- Quiz 3 information: Next Tuesday.
- Quiz 4 Thursday, November 29.
- Possible material for Quiz 4: Branch and Bound through Minimum Spanning Tree and Shortest Route Algorithms. Material from today's lecture and the November 27 lecture will **not** be on Quiz 4 (but will be on final).
- Network problems: 2007: Homework 3, #1; Final: #5,6 2008: Homework 3 # 1, 2; Final: #4.
- Next Tuesday is last day for midterm regrade requests.

Maximum Network Flow

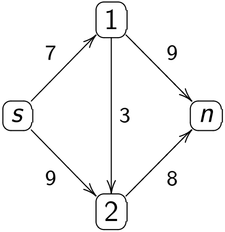
- Given: a network that consists of two distinguished nodes: the starting point or source and the end point, or sink.
- s denotes the source and n the sink.
- Edges are directed and have capacities.
- The maximum flow problem is to specify a nonnegative "load" to be carried on each edge that maximizes the total amount that reaches the sink subject to the constraint that no edge carries a load greater than its capacity.

Special Property

The maximum flow problem is an integer linear programming problem with the property that the solution to the relaxed problem (without integer constraints) will also solve the integer version of the problem. The special structure of the problem allows you to solve it using a simple algorithm.

Example

- There are two nodes in addition to the source and the sink.
- The capacity of the flow from (s) to (1) is 7.



Algorithm

1. Begin with no flows.
2. At each iteration you attempt to find an "augmenting path of labeled nodes" from (s) to (n).
3. You use the path to increase flow.
4. You continue this procedure until you cannot find a path of labeled nodes from (s) to (n). In order to describe the algorithm more completely, I must tell you how to label a path.

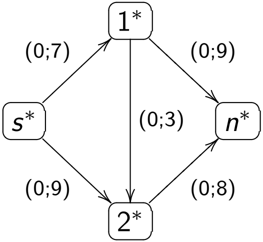
Rules

1. (s) is always labeled.
2. If Node (i) is labeled, then you can use it to label Node (j) if either:
   2.1 there exists an arc $(i) \rightarrow (j)$ with excess capacity or
   2.2 there is an arc $(j) \rightarrow (i)$ with positive flow.

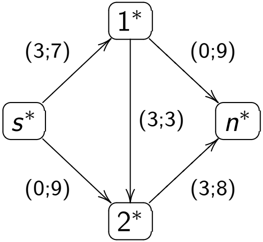
Example

- I put two numbers on each arc.
- The first represents the current flow.
- The second represents the arc capacity.
- A star (*) indicates that the node is labeled.

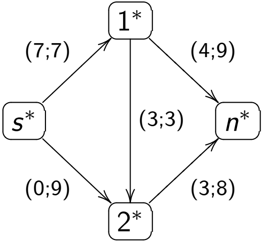
- Start with no flow.
- Construct an augmenting path.
  Example: $(s) \to (1) \to (2) \to (n)$. Put three units on this path (because three is the minimum of the used capacities).



- The next diagram includes three units shipped by the path found in Step 1.
- Another augmenting path is $(s) \to (1) \to (n)$.
- Put four units on this path. (More would violate the capacity constraint on $(s) \to (1)$.)

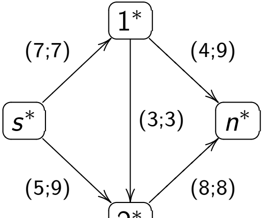


- Now $(s) \rightarrow (1)$ has no excess capacity.
- Put five units on $(s) \rightarrow (2) \rightarrow (n)$.



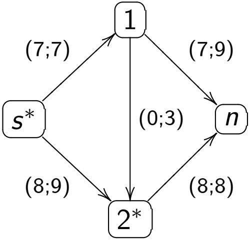
Flow Reversal

- The network below includes all flows constructed so far.
- I cannot label (1) directly because there is no excess capacity on $(s) \rightarrow (1)$.
- I can label (2).
- Furthermore, once (2) has a label, I can label (1) because $(1) \rightarrow (2)$ has positive flow.
- (n) can receive a label because $(1) \rightarrow (n)$ has excess capacity and (1) is labeled.
- The most I can put on the $(s) \rightarrow (2) \rightarrow (1) \rightarrow (n)$ route is three since three is the flow from $(1) \rightarrow (2)$.



Discussion

- When we add the three units we obtain the next diagram.
- This is the final step.
- We can label (s) and (2) but no other nodes.
- Hence it is not possible to find an augmenting path from (s) to (n).
- The diagram indicates the optimal flow.
- The total that can reach the sink is 15 (add up the amounts shipped on all of the nodes that reach (n) directly.
- In this case (1) $\to$ (n) and (2) $\to$ (n).



Obviously

Alternate Solution:

- First increase the flow to seven by using the path $(s) \rightarrow (1) \rightarrow (n)$.
- Next, increase the flow to fifteen by using the path $(s) \rightarrow (2) \rightarrow (n)$.

Why do it the Hard Way?

- Illustrate the possibility of labeling through "backwards" arcs as in Step 4.
- Demonstrate that the procedure will work to produce a solution no matter what order you generate augmenting paths.

Observation

When the given capacities are integers, the algorithm is guaranteed to finish in a finite number of steps and provide an answer that is also an integer.